*#import all the required libraries*

**import** seaborn **as** sn

**import** pandas **as** pd

**import** numpy **as** np

**import** random **as** rd

**import** statistics **as** st

**import** matplotlib.pyplot **as** plt

**import** sklearn **as** sk

In [93]:

*#Get the data from local repository*

path **=** "/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Assignment2/diabetes.csv"

df\_raw **=** pd**.**read\_csv(path)

In [94]:

*#show first 5 rows of dataset*

df\_raw**.**head()

Out[94]:

|  | **Pregnancies** | **Glucose** | **BloodPressure** | **SkinThickness** | **Insulin** | **BMI** | **DiabetesPedigreeFunction** | **Age** | **Outcome** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |

print("Number of Rows: {}\nNumber of Columns: {}"**.**format(df\_raw**.**shape[0], df\_raw**.**shape[1]))

Number of Rows: 768

Number of Columns: 9

*#let us check if we have any null values in the data*

df\_raw**.**isnull()**.**sum()

Pregnancies 0

Glucose 0

BloodPressure 0

SkinThickness 0

Insulin 0

BMI 0

DiabetesPedigreeFunction 0

Age 0

Outcome 0

dtype: int64

*#The data looks clean, so we can use the same raw data for our future analysis*

df\_raw**.**to\_csv('/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/CleanData/clean\_data.csv')

*##Let us now create a sample from the polulation*

*#we make use of random state to reproduce the result*

*#seed = random.seed(10)*

df\_getsample\_frompopulation**=** df\_raw**.**sample(n**=** 25, random\_state**=** 42)

*#fetch the first five rows of sample*

df\_getsample\_frompopulation**.**head()

Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome

668 6 98 58 33 190 34.0 0.430 43 0

324 2 112 75 32 0 35.7 0.148 21 0

624 2 108 64 0 0 30.8 0.158 21 0

690 8 107 80 0 0 24.6 0.856 34 0

473 7 136 90 0 0 29.9 0.210 50 0

*# number of rows are 25 in the smaple*

print("Number of Rows: {}\nNumber of Columns: {}"**.**format(df\_getsample\_frompopulation**.**shape[0], df\_raw**.**shape[1]))

Number of Rows: 25

Number of Columns: 9

In [99]:

*#find the mean Glucose of the sample on importing statistics package*

mean\_glucose\_sample **=** st**.**mean(df\_getsample\_frompopulation**.**loc[:,"Glucose"])

print(mean\_glucose\_sample)

116.64

In [101]:

*#find the max Glucose of the sample*

max\_glucose\_sample **=** max(df\_getsample\_frompopulation**.**loc[:,"Glucose"])

print(max\_glucose\_sample)

183

In [102]:

*#find the mean Glucose of the population*

mean\_glucose\_population **=** st**.**mean(df\_raw**.**loc[:,"Glucose"])

print(mean\_glucose\_population)

120.89453125

*#find the max Glucose of the population*

max\_glucose\_population **=** max(df\_raw**.**loc[:,"Glucose"])

print(max\_glucose\_population)

199

In [153]:

*# let us create a dictonary which stores the mean values of sample and population*

*# which helps in plotting the graphs*

**class** mean\_dictionary(dict):

**def** \_\_init\_\_(self):

self **=** dict()

**def** add(self, key, value):

self[key] **=** value

*# Main Function*

dict\_mean **=** mean\_dictionary()

dict\_mean**.**add('mean\_sample', mean\_glucose\_sample)

dict\_mean**.**add('mean\_population', mean\_glucose\_population)

print(dict\_mean)

*## Saving the results into results folder*

result\_save **=** open("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/A/dict\_mean.txt","w")

*# write file*

result\_save**.**write(str(dict\_mean))

*# close file*

result\_save**.**close()

{'mean\_sample': 116.64, 'mean\_population': 120.89453125}

In [154]:

*# let us create a dictonary which stores the max values of sample and population*

*# which helps in plotting the graphs*

**class** max\_dictionary(dict):

**def** \_\_init\_\_(self):

self **=** dict()

**def** add(self, key, value):

self[key] **=** value

*# Main Function*

dict\_max **=** max\_dictionary()

dict\_max**.**add('max\_sample', max\_glucose\_sample)

dict\_max**.**add('max\_population', max\_glucose\_population)

print(dict\_max)

*## Saving the results into results folder*

result\_save **=** open("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/A/dict\_max.txt","w")

*# write file*

result\_save**.**write(str(dict\_max))

*# close file*

result\_save**.**close()

{'max\_sample': 183, 'max\_population': 199}

*#creating the bar graph for sample vs population mean*

*# the graph clearly shows that mean\_population is slightly higher on copmaring with mean\_sample*

mean\_values **=** list(dict\_mean**.**keys())

values **=** list(dict\_mean**.**values())

fig **=** plt**.**figure(figsize **=** (4, 5))

*# creating the bar plot*

plt**.**bar(mean\_values, values, color **=**'blue',

width **=** 0.4)

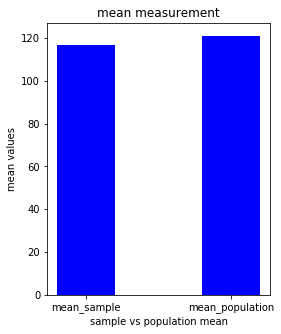
plt**.**xlabel("sample vs population mean")

plt**.**ylabel("mean values")

plt**.**title("mean measurement")

plt**.**savefig("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/A/mean\_barimage.jpeg")

plt**.**show()

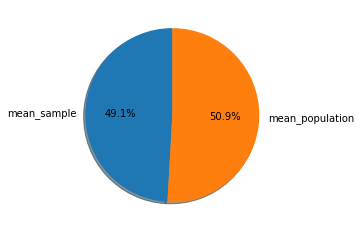


*#creating the pie graph for sample vs population mean*

*# the graph clearly shows that mean\_population is slightly higher on comparing with mean\_sample*

plt**.**pie(values, labels **=** mean\_values, autopct**=**'%1.1f%%', shadow**=True**, startangle**=**90)

plt**.**savefig("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/A/mean\_pieimage.jpeg")



*#creating the bar graph for sample vs population max*

*#the graph clearly shows that max\_population is higher on comparing with max\_sample*

max\_values **=** list(dict\_max**.**keys())

values **=** list(dict\_max**.**values())

fig **=** plt**.**figure(figsize **=** (4, 5))

*# creating the bar plot*

plt**.**bar(max\_values, values, color **=**'red',

width **=** 0.4)

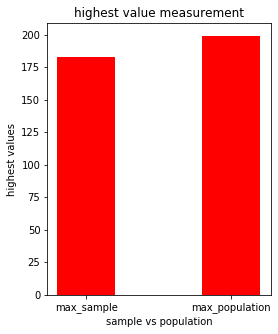
plt**.**xlabel("sample vs population")

plt**.**ylabel("highest values")

plt**.**title("highest value measurement")

plt**.**savefig("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/A/max\_barimage.jpeg")

plt**.**show()



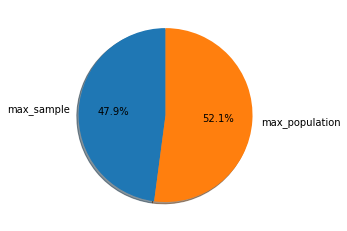
In [161]:

*#creating the pie graph for sample vs population max*

*#the graph clearly shows that max\_population is higher on comparing with max\_sample*

plt**.**pie(values, labels **=** max\_values , autopct**=**'%1.1f%%', shadow**=True**, startangle**=**90)

plt**.**savefig("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/A/max\_pieimage.jpeg")



*#let us now find the 98th percentile of sample data*

sample\_percentile **=** np**.**percentile(df\_getsample\_frompopulation**.**loc[:,"BMI"], 98)

print(sample\_percentile)

40.248

In [109]:

*#find the 98th percentile of population data*

population\_percentile **=** np**.**percentile(df\_raw**.**loc[:,"BMI"], 98)

print(population\_percentile)

47.52599999999996

In [162]:

*# let us create a dictonary which stores the percentile values of sample and population*

*# which helps in plotting the graphs*

**class** percentile\_dictionary(dict):

**def** \_\_init\_\_(self):

self **=** dict()

**def** add(self, key, value):

self[key] **=** value

*# Main Function*

dict\_percentile **=** percentile\_dictionary()

dict\_percentile**.**add('sample\_percentile', sample\_percentile)

dict\_percentile**.**add('population\_percentile', population\_percentile)

print(dict\_percentile)

*## Saving the results into results folder*

result\_save **=** open("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/B/dict\_percentile.txt","w")

*# write file*

result\_save**.**write(str(dict\_percentile))

*# close file*

result\_save**.**close()

{'sample\_percentile': 40.248, 'population\_percentile': 47.52599999999996}

In [163]:

*#creating the bar graph for sample vs population percentile*

*#the graph clearly shows that population\_percentile is higher on comparing with sample\_percentile*

percentile\_values **=** list(dict\_percentile**.**keys())

values **=** list(dict\_percentile**.**values())

fig **=** plt**.**figure(figsize **=** (4, 5))

*# creating the bar plot*

plt**.**bar(percentile\_values, values, color **=**'green',

width **=** 0.4)

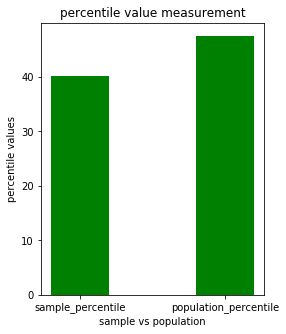
plt**.**xlabel("sample vs population")

plt**.**ylabel("percentile values")

plt**.**title("percentile value measurement")

plt**.**savefig("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/B/percentile\_barimage.jpeg")

plt**.**show()



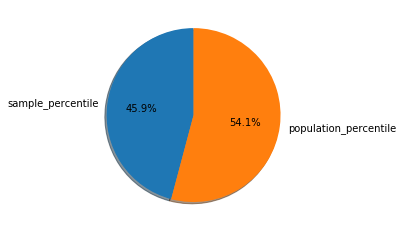
In [164]:

*#creating the pie graph for sample vs population percentile*

*#the graph clearly shows that population\_percentile is higher on comparing with sample\_percentile*

plt**.**pie(values, labels **=** percentile\_values , autopct**=**'%1.1f%%', shadow**=True**, startangle**=**90)

plt**.**savefig("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/B/percentile\_pieimage.jpeg")



*#Using bootstrap (replace= True), created 500 samples (of 150 observation each) from the population(raw\_data)*

boot **=** sk**.**utils**.**resample(df\_raw, replace**=True**, n\_samples**=**500, random\_state**=**150)

print(boot)

Pregnancies Glucose BloodPressure SkinThickness Insulin BMI \

228 4 197 70 39 744 36.7

442 4 117 64 27 120 33.2

507 1 130 60 23 170 28.6

25 10 125 70 26 115 31.1

496 5 110 68 0 0 26.0

.. ... ... ... ... ... ...

732 2 174 88 37 120 44.5

514 3 99 54 19 86 25.6

750 4 136 70 0 0 31.2

364 4 147 74 25 293 34.9

485 0 135 68 42 250 42.3

DiabetesPedigreeFunction Age Outcome

228 2.329 31 0

442 0.230 24 0

507 0.692 21 0

25 0.205 41 1

496 0.292 30 0

.. ... ... ...

732 0.646 24 1

514 0.154 24 0

750 1.182 22 1

364 0.385 30 0

485 0.365 24 1

[500 rows x 9 columns]

In [113]:

*#find the mean BMI of the bootstrapsample*

mean\_bootBMI **=** mean(boot**.**loc[:,"BMI"])

print(mean\_bootBMI)

31.8188

In [114]:

*#find the std BMI of the bootstrapsample*

std\_bootBMI **=** st**.**stdev(boot**.**loc[:,"BMI"])

print(std\_bootBMI)

8.058660594736716

In [115]:

*#find the percentile BMI of the bootstrapsample*

percentile\_bootBMI **=** np**.**percentile(boot**.**loc[:,"BMI"], 98)

print(percentile\_bootBMI)

46.50599999999999

In [116]:

*#find the mean BMI of the population*

mean\_rawBMI **=** mean(df\_raw**.**loc[:,"BMI"])

print(mean\_rawBMI)

31.992578125

In [117]:

*#find the std BMI of the population*

std\_rawBMI **=** st**.**stdev(df\_raw**.**loc[:,"BMI"])

print(std\_rawBMI)

7.884160320375446

In [119]:

*#find the percentile BMI of the population*

percentile\_rawBMI **=** np**.**percentile(df\_raw**.**loc[:,"BMI"], 98)

print(percentile\_rawBMI)

47.52599999999996

*# let us create a dictonary which stores the mean values of sample and population*

*# which helps in plotting the graphs*

**class** Bootmean\_dictionary(dict):

**def** \_\_init\_\_(self):

self **=** dict()

**def** add(self, key, value):

self[key] **=** value

*# Main Function*

dict\_Bootmean **=** Bootmean\_dictionary()

dict\_Bootmean**.**add('mean\_Bootsample', mean\_bootBMI)

dict\_Bootmean**.**add('mean\_population', mean\_rawBMI)

print(dict\_Bootmean)

*## Saving the results into results folder*

result\_save **=** open("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/C/dict\_Bootmean.txt","w")

*# write file*

result\_save**.**write(str(dict\_Bootmean))

*# close file*

result\_save**.**close()

{'mean\_Bootsample': 31.8188, 'mean\_population': 31.992578125}

*#creating the bar graph for sample vs population mean*

*#the graph clearly shows that population is higher on comparing with sample*

Bootmean\_values **=** list(dict\_Bootmean**.**keys())

values **=** list(dict\_Bootmean**.**values())

fig **=** plt**.**figure(figsize **=** (4, 5))

*# creating the bar plot*

plt**.**bar(Bootmean\_values, values, color **=**'orange',

width **=** 0.4)

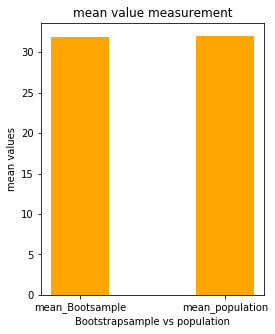
plt**.**xlabel("Bootstrapsample vs population")

plt**.**ylabel("mean values")

plt**.**title("mean value measurement")

plt**.**savefig("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/C/bootmean\_barimage.jpeg")

plt**.**show()

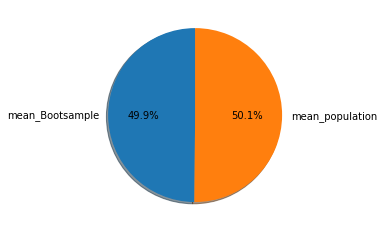


*#creating the pie graph for sample vs population mean*

*#the graph clearly shows that population is higher on comparing with sample*

plt**.**pie(values, labels **=** Bootmean\_values, autopct**=**'%1.1f%%', shadow**=True**, startangle**=**90)

plt**.**savefig("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/C/bootmean\_pieimage.jpeg")



In [168]:

*# let us create a dictonary which stores the std values of sample and population*

*# which helps in plotting the graphs*

**class** Bootstd\_dictionary(dict):

**def** \_\_init\_\_(self):

self **=** dict()

**def** add(self, key, value):

self[key] **=** value

*# Main Function*

dict\_Bootstd **=** Bootstd\_dictionary()

dict\_Bootstd**.**add('std\_Bootsample', std\_bootBMI)

dict\_Bootstd**.**add('std\_population', std\_rawBMI)

print(dict\_Bootstd)

*## Saving the results into results folder*

result\_save **=** open("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/C/dict\_Bootstd.txt","w")

*# write file*

result\_save**.**write(str(dict\_Bootstd))

*# close file*

result\_save**.**close()

{'std\_Bootsample': 8.058660594736716, 'std\_population': 7.884160320375446}

In [170]:

*#creating the bar graph for sample vs population std*

*#the graph clearly shows that population is slightly higher on comparing with sample*

Bootstd\_values **=** list(dict\_Bootstd**.**keys())

values **=** list(dict\_Bootstd**.**values())

fig **=** plt**.**figure(figsize **=** (4, 5))

*# creating the bar plot*

plt**.**bar(Bootstd\_values, values, color **=**'silver',

width **=** 0.4)

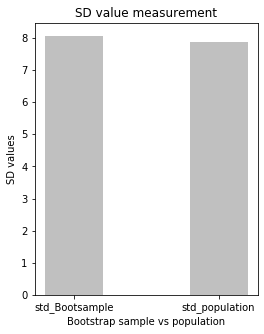
plt**.**xlabel("Bootstrap sample vs population")

plt**.**ylabel("SD values")

plt**.**title("SD value measurement")

plt**.**savefig("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/C/bootstd\_barimage.jpeg")

plt**.**show()



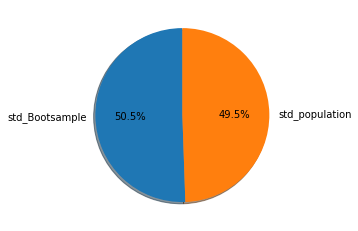
In [171]:

*#creating the pie graph for sample vs population std*

*#the graph clearly shows that population is slightly higher on comparing with sample*

plt**.**pie(values, labels **=** Bootstd\_values , autopct**=**'%1.1f%%', shadow**=True**, startangle**=**90)

plt**.**savefig("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/C/bootstd\_pieimage.jpeg")



*# let us create a dictonary which stores the percentile values of sample and population*

*# which helps in plotting the graphs*

**class** Bootpercentile\_dictionary(dict):

**def** \_\_init\_\_(self):

self **=** dict()

**def** add(self, key, value):

self[key] **=** value

*# Main Function*

dict\_Bootpercentile **=** Bootpercentile\_dictionary()

dict\_Bootpercentile**.**add('percentile\_Bootsample', percentile\_bootBMI)

dict\_Bootpercentile**.**add('percentile\_population', percentile\_rawBMI)

print(dict\_Bootpercentile)

*## Saving the results into results folder*

result\_save **=** open("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/C/dict\_Bootpercentile.txt","w")

*# write file*

result\_save**.**write(str(dict\_Bootstd))

*# close file*

result\_save**.**close()

{'percentile\_Bootsample': 46.50599999999999, 'percentile\_population': 47.52599999999996}

*#creating the bar graph for sample vs population percentile*

*#the graph clearly shows that population is slightly higher on comparing with sample*

Bootpercentile\_values **=** list(dict\_Bootpercentile**.**keys())

values **=** list(dict\_Bootpercentile**.**values())

fig **=** plt**.**figure(figsize **=** (4, 5))

*# creating the bar plot*

plt**.**bar(Bootpercentile\_values, values, color **=**'purple',

width **=** 0.4)

plt**.**xlabel("Bootstrap sample vs population")

plt**.**ylabel("Percentile values")

plt**.**title("percentile value measurement")

plt**.**savefig("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/C/bootpercentile\_barimage.jpeg")

plt**.**show()



In [174]:

*#creating the bar graph for sample vs population percentile*

*#the graph clearly shows that population is slightly higher on comparing with sample*

plt**.**pie(values, labels **=** Bootpercentile\_values, autopct**=**'%1.1f%%', shadow**=True**, startangle**=**90)

plt**.**savefig("/Users/aditya/Documents/My-files/Umkc/Spring/spring23/PDS/Project/Diabetes/Results/C/bootpercentile\_barimage.jpeg")

